**Программа учебной дисциплины «Алгоритмы и структуры данных» (часть 2)**

Утверждена

Академическим советом ООП

Протокол № от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

|  |  |
| --- | --- |
| Автор(ы)  | Дегтярев К. Ю., к.т.н., доцент, Шершаков С. А., ст. преподаватель, Дворянский Л.В., ст. преподаватель |
| Число кредитов  | 8 |
| Контактная работа (час.)  | 84 |
| Самостоятельная работа (час.)  | 80 |
| Курс  | 2 |
| Формат изучения дисциплины | без обязательного использования он-лайн курса(-ов) |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

 Изучение данной дисциплины базируется на знаниях, полученных студентами при освоении учебных дисциплин «Дискретная математика», «Программирование», «Алгорит-мы и структуры данных» (часть 1) первого и второго годов обучения, основ программи-рования в части базовых алгоритмических конструкций. Таким образом, «Алгоритмы и структуры данных» (часть 2) можно рассматривать в качестве одной из важных состав-ляющих цепочки предлагаемых в рамках БУП направления **09.03.04** дисциплин, связан-ных с теоретической информатикой, основами информационных технологий и програм-мирования (первых двух лет обучения).

 Дисциплина является основой для последующего изучения таких дисциплин как «Проектирование и архитектура программных систем», «Обеспечение качества и тестиро-вание» (3-й курс), отдельных научно-исследовательских семинаров (3-й и 4-й курсы | цикл Б.6 ‘Научный семинар’), а также выполнения курсовых, проектных и выпускных работ, предусмотренных БУП по направлению подготовки 09.03.04.

 Целями освоения дисциплины «Алгоритмы и структуры данных» (часть 2) являются

* формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области структур данных и теории алгоритмов, пониманием концепции абстрактных типов данных (АТД) и подходов к их реализации на языке С++ на основе принципов процедурного и объектно-ориентированного построения программ, оценки влияния выбора структур данных и/или алгоритмов на производительность (time) и пространственную (space) эффек-тивность программ;
* получение практических навыков решения задач с использованием разных структур данных (напр., линейных списков, стеков, деков, очередей, хэш-таблиц и пр.), используя концепции абстракции данных и модульного программирования;
* развитие умений, основанных на полученных теоретических знаниях, позволяющих на творческом и репродуктивном уровне предлагать и применять эффективные подходы к решению (алгоритмизации) поставленных задач с использованием данных простой и сложной структуры;
* получение студентами навыков самостоятельной исследовательской работы, предполагающей изучение специфических особенностей работы со структурами данных в рамках разработки подходов (алгоритмов) к решению поставленной задачи, вопросов управления памятью в С++ и использования компонентов стандартной библиотеки шаблонов (STL).

В результате освоения дисциплины студент должен:

**знать:**

* базовые абстрактные типы данных (АТД), понимать их особенности, применимые операции и методы реализации АТД на языке С++ (структура данных как конкретная форма реализации АТД); контейнеры, итераторы и алгоритмы как основные компоненты STL,

**уметь:**

* разрабатывать С++ программы, реализующие заданный алгоритм и использующий определенные структуры данных,

**владеть:**

* навыками и основами процедурного и объектно-ориентированного программирования на языке С++, работы с шаблонами функций и классов, использования STL как реализованную на языке С++ коллекцию обобщенных (generic) структур данных (контейнеров) и алгоритмов, методами оценки трудоемкости (сложности) алгоритмов, подходами к измерению времени в программных реализациях алгоритмов решения поставленных задач.
1. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Тема 1.** Вводная часть. Понятие абстрактного типа данных (АТД). Типы данных, структуры данных и алгоритмы. Реализация АТД. Указатели. Объектно-ориентированное программирование и С++. Шаблоны, классы.

**Тема 2.** Структуры данных и стандартная библиотека шаблонов (STL) – введение. Классы и объекты в языке С++. Конструкторы, деструктор (особенности и ключевые моменты). Перегрузка операций.

**Тема 3.** Интеллектуальные указатели (smart pointers). Динамическое выделение памяти. АТД "Список" (List). Связные структуры (списки). Одно- и двусвязные списки. Элементы управления памятью.

**Тема 4.** Ссылочные переменные. Контейнеры (последовательные, ассоциативные, неупорядоченные). Итераторы. Категории итераторов. Алгоритмы. Контейнеры STL. Последовательные контейнеры. Массивы. Векторы. Векторы и списки в STL.

**Тема 5.** Стеки, очереди и деки. АТД "Стек" (Stack) и "Очередь" (Queue). Пользовательские классы стека и очереди (реализация стека и очереди). Приоритетная очередь (куча).

**Тема 6.** Рекурсия. Вызовы функций (системный стек, запись активации).

**Тема 7.** Контейнеры STL (списки) и алгоритмы STL. Строки. АТД "Строка". Класс std::string (реализация). Описание строковых классов (типы и операции).

**Тема 8.** Двоичные деревья. Представление двоичных деревьев. АТД "Дерево поиска" – деревья двоичного поиска. Красно-черные деревья, AVL-деревья.

**Тема 9.** Списки. Множества и отображения в стандартной библиотеке.

**Тема 10.** Хэширование. Разрешение коллизий. Хэш-таблицы в стандартной библиотеке. Качество хэш-функции. Заполнение хэштаблицы значениями с использованием заданных хэш-функции и подходов.

**Тема 11.** Алгоритмы (модифицирующие и немодифицирующие, сортировки, удаления, перестановки). Дополнительные комментарии, заключение Анализ рекурсивных алгоритмов методом прямого подсчета вершин рекурсивного дерева. Пример анализа – алгоритм сортировки слиянием.

**Примерный перечень тем практических занятий:**

1. Простейший пример класса и работа с объектами – класс Point (точка в двумерном евклидовом пространстве);
2. Работа со стеком (Stack) и безопасным массивом (Safe Array);
3. Работа со связным списком (Linked List – BrokenDNA);
4. Усложненный вариант работы сос связными списками;
5. Реализация очереди с использованием стеков (StackQueue);
6. Работа с очередью – Printer Simulation (FIFO);
7. Работа со SkipList;
8. Хэширование – SpellChecking;
9. Работа с В-деревьями;
10. Работа с AVL и красно-черными деревьми.

На практических занятиях языком программной реализации алгоритмов является язык программирования С++, в связи с этим на занятиях изучаются основы языка С++ в объеме, необходимом для выполнения практических заданий.

1. **ОЦЕНИВАНИЕ**

**Формы контроля знаний студентов**



**3.1 Критерии оценки знаний, навыков**

Оценки по всем формам контроля выставляются по 10-ти балльной шкале.

**Текущий контроль** предусматривает одну контрольную работу (КР) и одно до-машнее задание (ДЗ), выполняемые в четвертом модуле.

 *Контрольная работа* предусматривает программную реализацию на компьютере пред-ложенного алгоритма; выполняется во время часов практического занятия. Оценка за кон-трольную работу выставляется c учетом критериев, описанных в разделе IV (контрольная работа) программы. Пересдача контрольной работы с целью повышения оценки не допус-кается. Допускается (в каждом конкретном случае решение принимается преподавателя-ми, ведущими занятия) выполнение контрольной работы в более поздний срок при про-пуске по особо уважительной причине (рассматривается в каждом конкретном случае).

 *Домашнее задание* (1 задание, состоящее из одной или нескольких частей) включает анализ, реализацию, тестирование и отладку программ(ы). По домашнему заданию оформляется отчет в электронном виде. Домашнее задание размещается в LMS в разделе «Проекты». В установленный и объявляемый заранее срок каждый студент загружает в LMS архив, содержащий полностью оформленный отчет и программу решения предло-женного домашнего задания. Оценка за домашнее задание выставляется с учетом полноты выполнения задания и оформления результатов, т.е. при выставлении оценки принимают-ся во внимание все аспекты выполненной студентом работы.

Кроме отмеченных выше форм текущего контроля также учитываются

• работа студентов на практических занятиях, а именно: реализация алгоритмов, решение тестов по некоторым темам,

• (опция) выполнение домашней работы (заданий к практическим занятиям).

Соответствующие задания размещаются в LMS и сдаются студентами в указанный срок в виде проектов. В случае несвоевременной сдачи домашних работ и домашнего задания оценка снижается на (1) один балл за каждый день задержки, при этом не допускается задержка более, чем на 4 дня. При задержке по уважительной причине баллы не снимаются.

 **Итоговый контроль:** экзамен в конце 4-го модуля (проводится в устной форме). Экзамен состоит из двух частей, а именно: (а) теоретической части, которая проводится в форме устной беседы по тематике курса «Алгоритмы и структуры данных» (часть 2) - (~30÷35 мин.), и (б) практическ. части, связанной с обсуждением результатов домашнего задания (~30÷35 мин.).

**3.2 Порядок формирования оценок по дисциплине**

По всем видам работ выставляется 10-балльная оценка.

**Oмодуль2** – промежуточная оценка, полученная студентом в конце 2-го модуля (эта оценка проставлена в ведомость по окончании экзамена по дисциплине «Алгоритмы и структуры данных» (часть 1) в конце декабря 2016 г.). Предусматривается оценивание работ студентов на практических занятиях **Опр-класс**; в зависимости от подхода к проведению практич. занятий по дисциплине в эту оценку могут входить

(а) результаты тестирования по текущей теме, (б) разработка (написание) программ (отдельных частей программ). Оценки за работу на практических занятиях выставляются в рабочую ведомость. Оценивается самостоятельная работа студентов **Опр-сам.работа**; в этой оценке учитывается оригинальность, правильность и полнота выполнения выданных для завершения в течение определенного времени заданий по темам практических занятий. Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость.

 Накопленная оценка за текущий контроль учитывает показанные студентом результаты следующим образом:

**Онакопленная** = 0,15\***Омодуль2** + 0,4\***Отекущий** + 0,2\***Опр-класс** + 0,25\***Опр-сам.работа**,

где оценка Отекущий рассчитывается как взвешенная сумма оценок за домашнее задание (ДЗ) и контрольную работу (КР), а именно: **Отекущий** = 0,5\***Одз** + 0,5\***Окр**. Накопленная оценка **Онакопленная** не округляется.

 Итоговая оценка (четвертый модуль) вычисляется на основе полученной оценки накопления **Онакопленная** и оценки за экзамен **Оэкзамен**, т.е. **Оитог** = 0,6\***Онакопленная** + 0,4\***Оэкзамен**. Итоговая оценка **Оитог** – результат арифметического округления. В случае получения неудовлетворительной оценки и повторной сдачи (пересдачи) экзамена формула вычисления оценки студента не меняется.

 Перевод в пятибалльную оценку осуществляется в соответст. со следующей таблицей.

**Таблица соответствия оценок по десятибалльной и пятибалльной системам**



1. **ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

# **Оценочные средства для текущего контроля студента**

**Домашнее задание**

Предусматривается подробный анализ, разработка алгоритма решения и его программная реализация применительно к предложенной проблеме (задаче), связанной с рассматриваемым в рамках данной дисциплины материалом. Должно быть выполнено планирование эксперимента, в т.ч. определение входных и выходных данных. Результаты сравнительного анализа должны быть оформлены в виде таблиц, графиков, текста.

**Контрольная работа**

Примерное задание (шаблон) описано в программе дисциплины «Алгоритмы и структуры данных» (часть 1). Там же приведено и краткое описание подхода к выставлению оценки за контрольную работу (оставлено без изменения), а именно:

1) Алгоритм реализован верно - 4 балла (из 10 возможных),

2) Верно выполняется тестовый пример, показанный в описании задания - 5 баллов,

3) Тестовый пример выполняется не на всех входных данных - (+ до 2 баллов) - 6-7 баллов.

4) Тестовый пример выполняется на всех входных данных (+ до 2 баллов) - 8-9 баллов.

Работа выполнена самостоятельно менее, чем за 60 минут — +1 балл к оценке, полученной в соответствии с пп. 1)-4).

**Оценочные средства для промежуточной аттестации**

Возможные вопросы для оценки качества освоения дисциплины

1. Как бы вы определили понятие «структура данных»?

2. Как бы вы определили понятие «абстракция данных»? Что такое «абстракция» и в чем состоит ее важность (значимость). Приведите пример абстрактного типа данных. Сформулируйте и кратко прокомментируйте преимущества использования АТД,

3. Что собой представляет динамическая структура данных?

4. Определите АТД для символьных строк (АТД должен включать типичные функции, которые применяются к строкам; каждую функцию определите в терминах входных и выходных данных). Предложите две разные реализации строкового типа,

5. Что такое дек? Что такое стек? Объясните, чем отличается стек от очереди. Как выглядит пользовательский класс стека, очереди, очереди с приоритетом (STL)?

6. Объясните отличие класса (class) от структуры (struct),

7. Какие из структур данных являются индексируемыми структурами?

8. Что такое связный список? Как можно представить разреженную матрицу (sparse matrix) с использованием связного списка? Как можно эффективно представить разреженную матрицу с использованием массива? Что можно сказать о сложности хранения (space complexity) такого подхода?

9. Приведите примеры использования векторов, деков, списков, множеств (мультимножеств) (STL),

10. Предположим, что текущая конфигурация очереди имеет вид: *a b c d*. Сколько понадобится операций извлечения/размещения для получения конфигурации *d c b a*?

11. Как организовать поиск заданного элемента в односвязном списке (SLL)?

12. Что собой представляет дерево бинарного поиска (binary search tree)?

13. Что собой представляют классы shared\_ptr и unique\_ptr (типы интеллектуальных указателей)? 14. Какой алгоритм сортировки считается самым быстрым?

15. Какие вы знаете типы контейнеров (STL)?

16. Какую работу выполняет хэш-функция?

1. **РЕСУРСЫ**

**5.1 Основная литература**

- Вирт, Никлаус. «Алгоритмы и структуры данных» (сер. Библиотека программиста), Пер. с англ. — СПб.: Невский Диалект, 2001 *(или более ранние + поздние издания; доступно в библиотеке)*,

- Кормен, Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 3-е издание. — Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2005. — 1328 с. ) *(или более поздние издания; доступно в библиотеке)*,

- Кнут, Дональд Э. «Искусство программирования, том 2. Получисленные алгоритмы», **3**-е издание, — Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2012. - 832 стр. *(или более поздние издания)*,

- Кнут, Дональд Э. «Искусство программирования, том 1. Основные алгоритмы», **3**-е издание,— Пер. с англ. — М.; СПб.; Киев: ООО «И.Д. Вильямс», 2011. - 720 стр. *(или более поздние издания; доступно в библиотеке)*,

- Гагарина, Л.Г., Колдаев, В.Д. Алгоритмы и структуры данных, М.: «Финансы и статистика» (Инфра-М), 2009, 303 с. *(дополнительно)*.

**5.2 Справочные материалы, словари, энциклопедии, инф. сайты** (*в свободном доступе*)

- *(на англ. языке)* Collected Algorithms (ACM), htpp://www.acm.org/calgo

- TutorialPoint: C++ STL Tutorial, <http://www.tutorialspoint.com/cplusplus/cpp_stl_tutorial.htm>

- Контейнеры STL (MSDN), <http://msdn.microsoft.com/ru-ru/library/1fe2x6kt.aspx>

- *(на русск. языке)* Справка по С++, <http://ru.cppreference.com/w/> | *(на англ. языке)* C++ Reference, <http://en.cppreference.com/w/>

- *(на англ. языке)* Standard C++ Library Reference, <http://www.cplusplus.com/reference/>

- *(на англ. языке)* Shaffer C.A. Data Structures and Algorithm Analysis (C++ version/3.2), Virginia Tech, 2010-13, people.cs.vt.edu/shaffer/Book/C++3elatest.pdf

- Using C++11’s Smart Pointers (Course EECS381 “Object-oriented and Advanced Programming Handouts, UM), 2014, [http://www.umich.edu/~eecs381/handouts/C++11\_smart\_ptrs.pdf](http://www.umich.edu/~eecs381/handouts/C%2B%2B11_smart_ptrs.pdf)

**5.3 Программное обеспечение**

 Практические занятия проводятся в компьютерных классах с выходом в Интернет и доступом к ресурсам электронной библиотеки (<http://library.hse.ru/e-resources/e-resources.htm> ) НИУ ВШЭ. Предусматривается наличие у каждого студента рабочего места. Практическая работа (практические занятия по дисциплине) ориентирована на использование современных интегрированных инструментальных сред разработки.

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|  1. |  Microsoft Windows 7 Professional RUSMicrosoft Windows 10 | *Из внутренней сети университета (договор)* |
|  2. | Microsoft Office Professional Plus 2010 | *Из внутренней сети университета (договор)* |
|  3. | Microsoft Visual Studio 2015 Community | *Свободно распространяемое ПО* |
|  4. | JetBrains CLion Community Edition | *Свободно распространяемое ПО* |
|  5. | GNU Compiler Collection (GCC) | *Свободно распространяемое ПО* |

* 1. **Материально-техническое обеспечение дисциплины**

 Учебные аудитории для лекционных занятий по дисциплине, оснащенные мультимедийным проектором с дистанционным управлением.

 Учебные аудитории для семинаров и самостоятельной работы, оснащенные компьютерами, на которых установлены интегрированные среды разработки Microsoft Visual Studio версии 2013-2015 (и выше) и/или CLion версии 2016.0 и выше, набор компиляторов GNU Compiler Collection (GCC) версии 6.1 и выше. Компьютеры подключены к сети Интернет для обеспечения доступа студентов к справочным материалам.